

## LOFAR Phase Self-Calibration Pipeline

*Nicolas CITT (calibration Imaging Tiger Team) workshop  
16 September 2015, ASTRON*

# Phase Self-Calibration concept

- **Goal:**

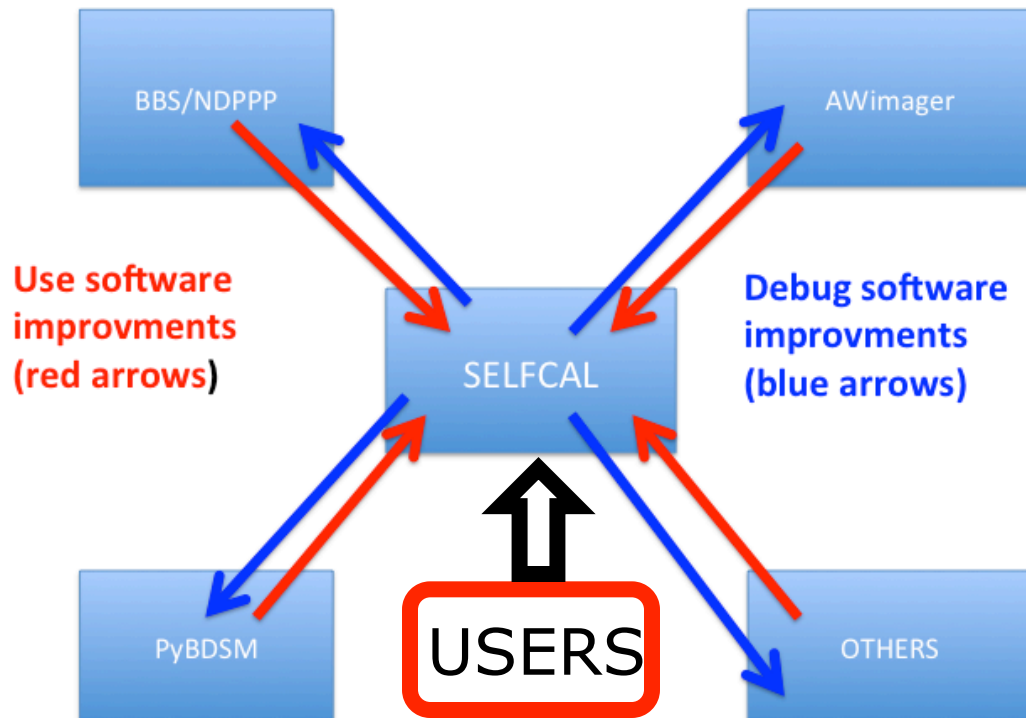
- Calibrate the Phase with the most accurate sky model

- **How to do:**

- Define cycle like:
    - Phase Calibration/flagging
    - Imaging
    - Source extraction => mask generation
    - Imaging with mask and use the extracted clean component Skymodel to calibrate the next cycle
- Increasing at each cycle the image resolution

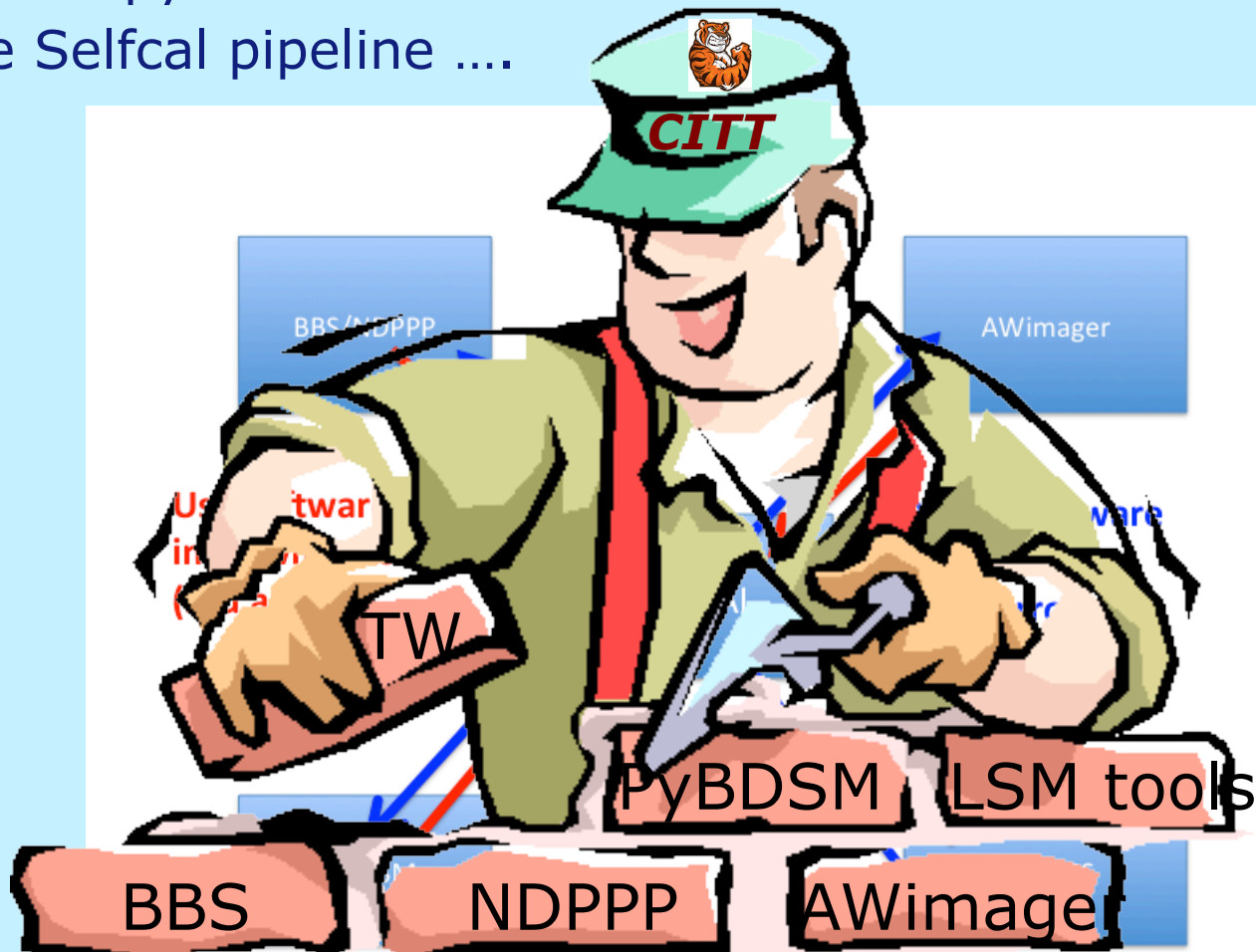
# Phase Self-Calibration concept

- In the context of the CITT:
  - Selfcal.py uses all resources of CITT as bricks to build the Selfcal pipeline ....



# Phase Self-Calibration concept

- **Building selfcal means to be a kind of bricklayer:**
  - Selfcal.py uses all resources of CITT as bricks to build the Selfcal pipeline ....



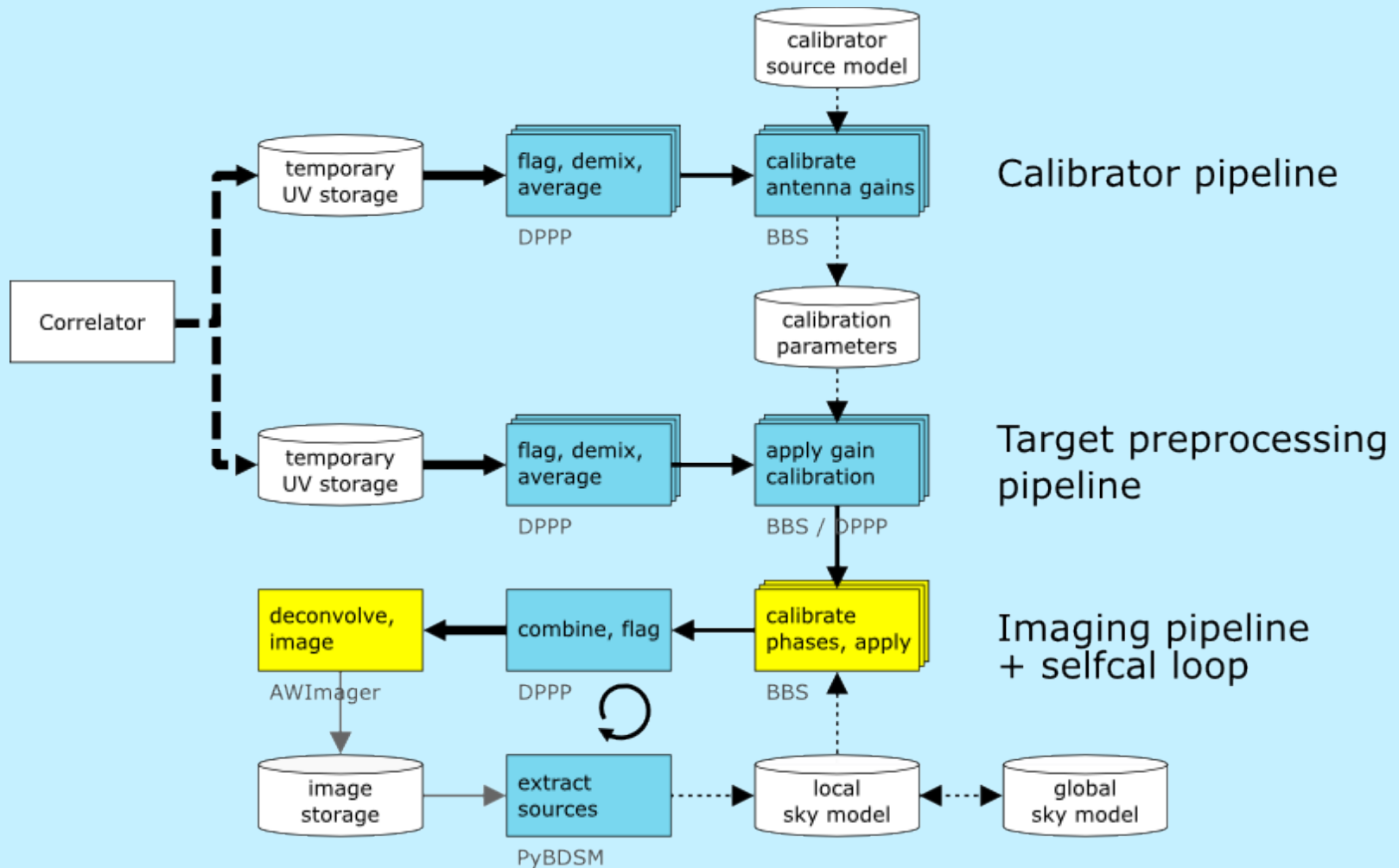
## 2 PLAYGROUNDS

```
graph TD; A[2 PLAYGROUNDS] --> B[Selfcal.py Standalone script]; A --> C[Selfcal Radio-Observatory Pipeline];
```

Selfcal.py  
Standalone script

Selfcal  
Radio-Observatory  
Pipeline

# LOFAR imaging pipeline: Overview



Selfcal version 1.0 implemented and in commissioning:

- Limited resolution to 20 arcsec
- Robust weighting parameter frozed at -0.3
- Resolution vector frozed (90'',70'',45'',30'',20'')

Framework troubles (communication between nodes)

**Hope ready for LOFAR cycle 5 (October 2015)**

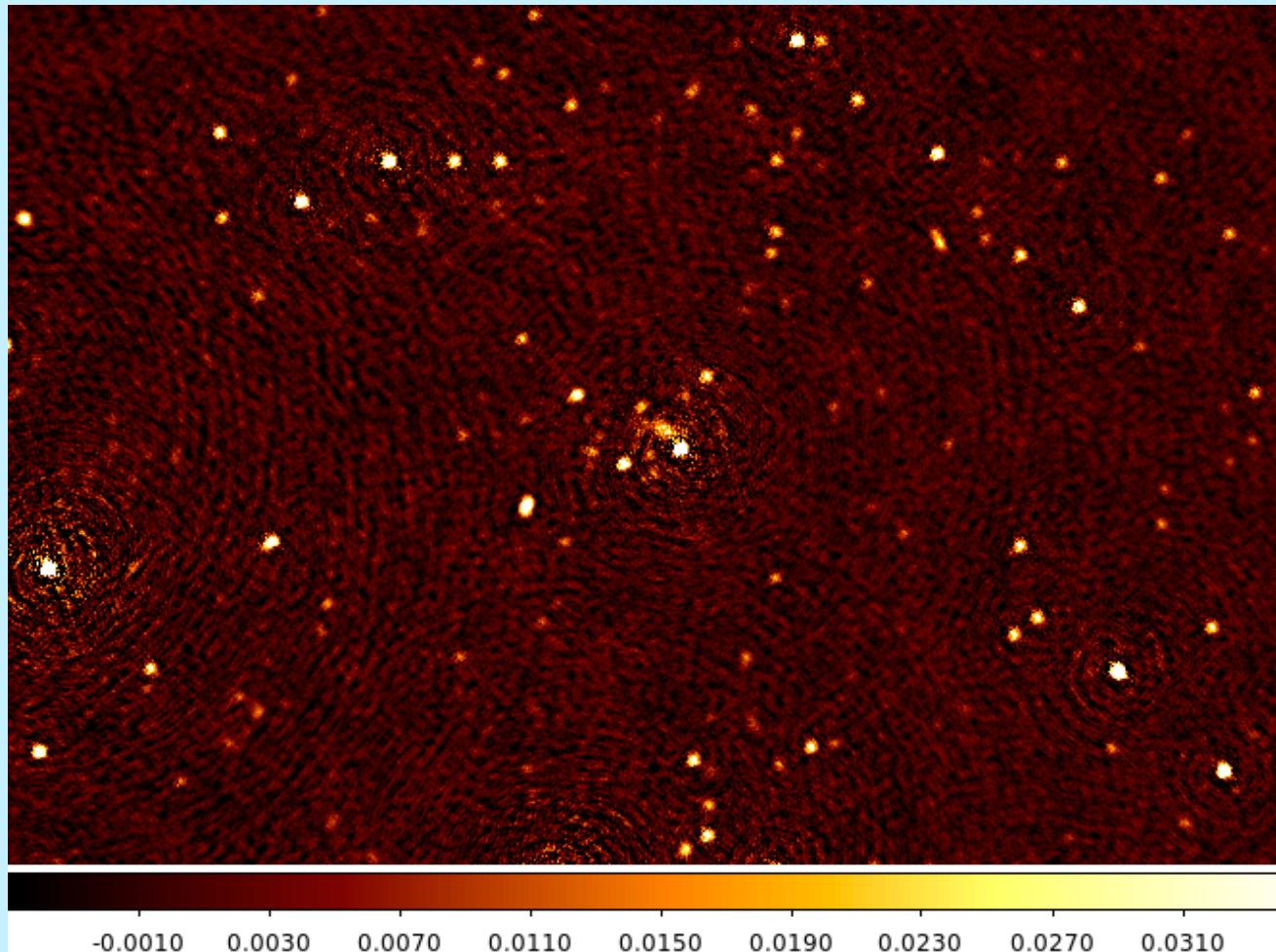


# Selfcal Radio-Observatory Pipeline

## Commissioning example:

ASTRON

~6h30 min observation; 10 Subbands (2 MHz bandwidth) at ~153 MHz;  
**rms=1.9 mJy/Beam**; DR~9450; resolution = 10 arcsec





# Selfcal.py Standalone script

- **How to get it:** a simple search on github “Pyselfcal”
- Git clone <https://github.com/nicolasvilchez/pyselfcal.git>
- Then cd pyselfcal
- **Python setup.py install --prefix=/yourpath/**
- (default is /usr/local, if --user specified it go to /home/user/.local/bin)
- On CEP just type:
- **use Pyselfcal**

- **Preprocessing:**

- Prepare your data: mergeSB.py=> subbands concatenation (increase the signal/noise) and select Nof channel/subband
  - Typically: 20 subbands and 1 channel/subband

- **Processing:**

- Requirements: Observation (only frequency concatenated data) and output directories
- Other parameters have default values
- Internal checks (frequency, nof subbands, fov, etc ...)

# Selfcal.py Standalone script

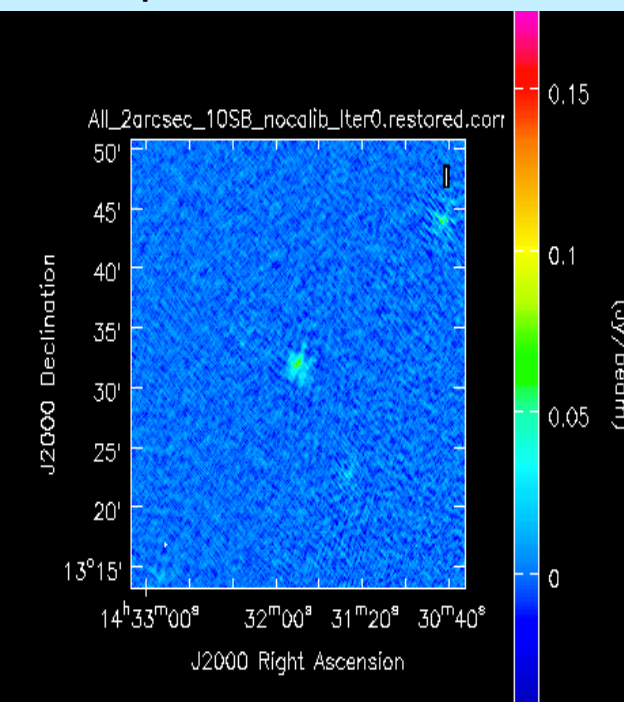
## ■ List of parameters (type selfcal.py -h in a terminal):

- \* --nbCycle(int, default:10):
- \* --skyModel(string,default:'')
- \* --outerfovclean(string,default:no):
- \* --VLSSuse(string,default:yes):
  
- \* --annulusRadius(float, default:1):
- \* --startResolution(float, default:15xbest resolution available in the data)
- \* --endResolution(float, default:best resolution available in the data)
- \* --resolutionVector(list of float: --resolutionVector=x1, x2, x3 (do not use parenthesis),default:[none])
- \* --startingFactor(int, default:15)
  
- \* --mask(string,default:yes):
- \* --maskDilation(integer,default:0)
- \* --UVmin(float, default:0 or 0.1 in klambda)
- \* --FOV(float, default:5 in degree)
- \* --nofPixelPerBeam(float, default:4)
- \* --robust(float, default:none)
- \* **--peeling(string, default:no) => currently in test**
- \* **--nofDirections=(int, default:1) => currently in test**

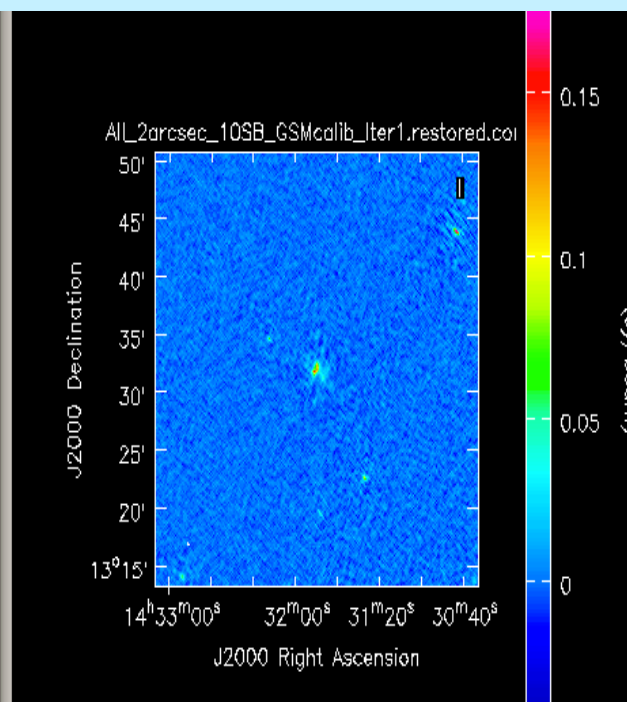
# Results on J1431+1331:

10 Subbands (2 MHz bandwidth) at  $\sim 153$  MHz; resolution = 5 arcsec  
 $\sim 8$ h observation

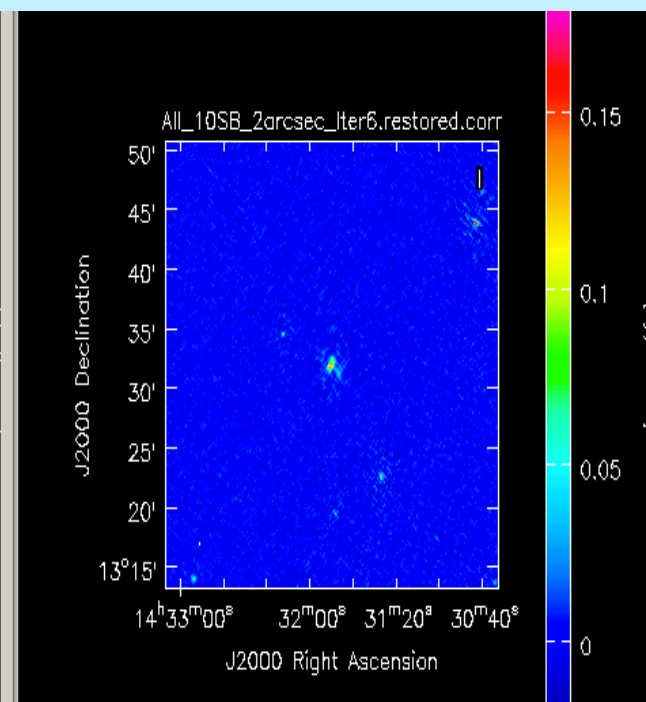
No phase Calibration



GSM Calibration



Self-Calibration



Noise  $\sim 25$  mJy

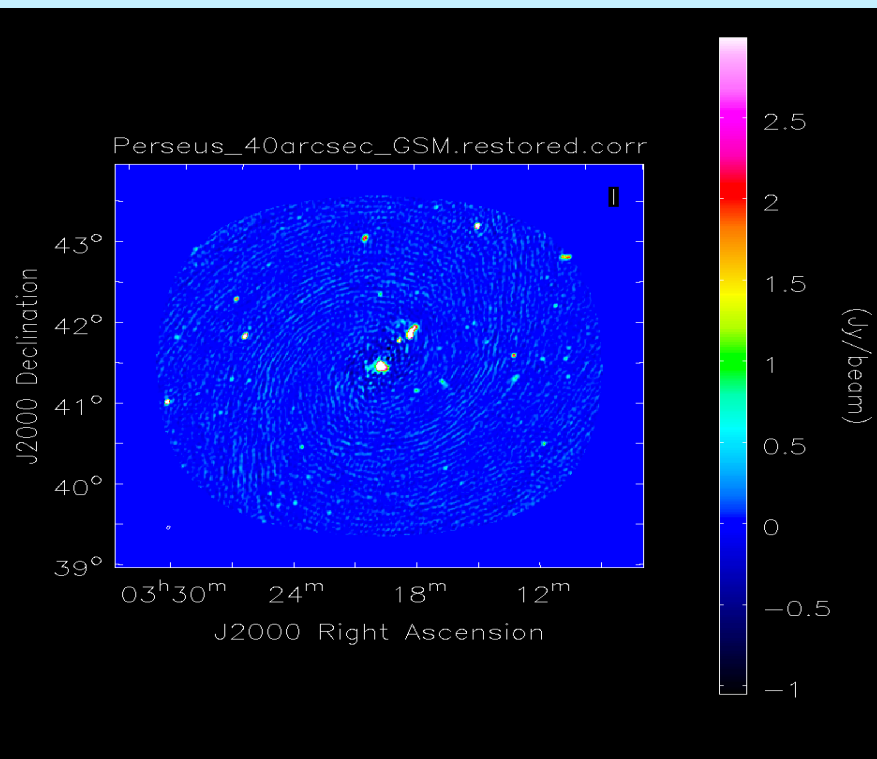
Noise  $\sim 10$  mJy

Noise  $\sim 1.5$  mJy

# Results on Perseus cluster:

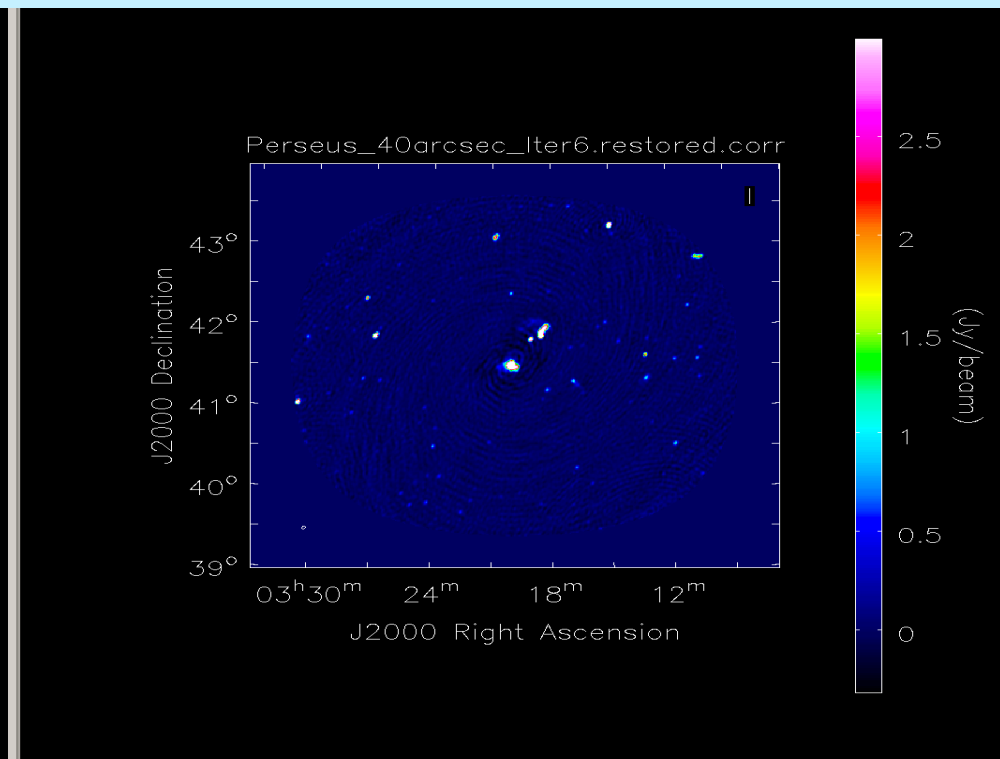
10 Subbands (2 MHz bandwidth) at  $\sim 170$  MHz; resolution = 25 arcsec  
 $\sim 9$ h30 observation

GSM calibration



Rms $\sim 80$  mJy

Self-calibration

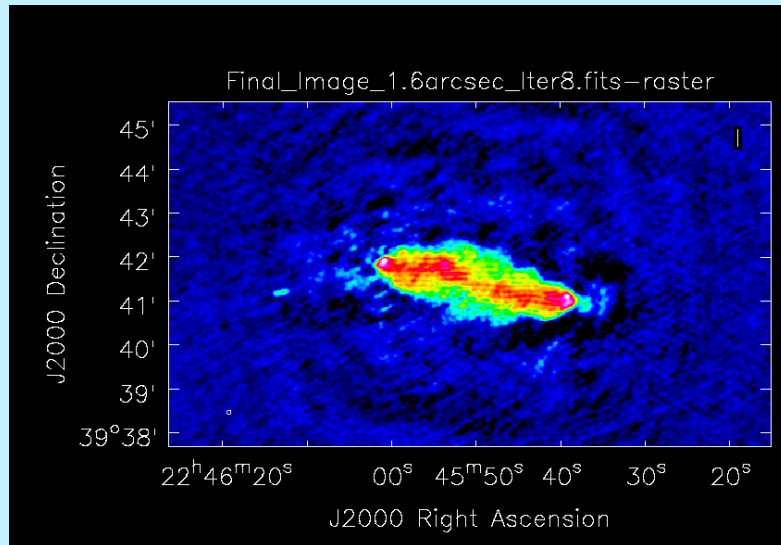


Rms $\sim 50$  mJy

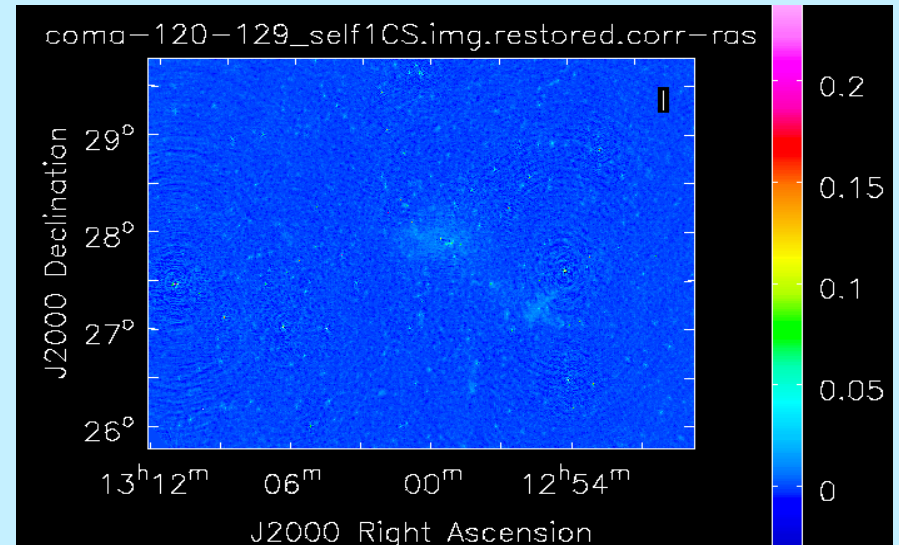
# Selfcal.py Standalone script

## Limitations:

4C452 => extended source  
requires multi-scale cleaning



Coma clusters => extended source  
requires multi-scale cleaning



Awimager 2.0 must be available soon ...



# Selfcal.py Standalone script

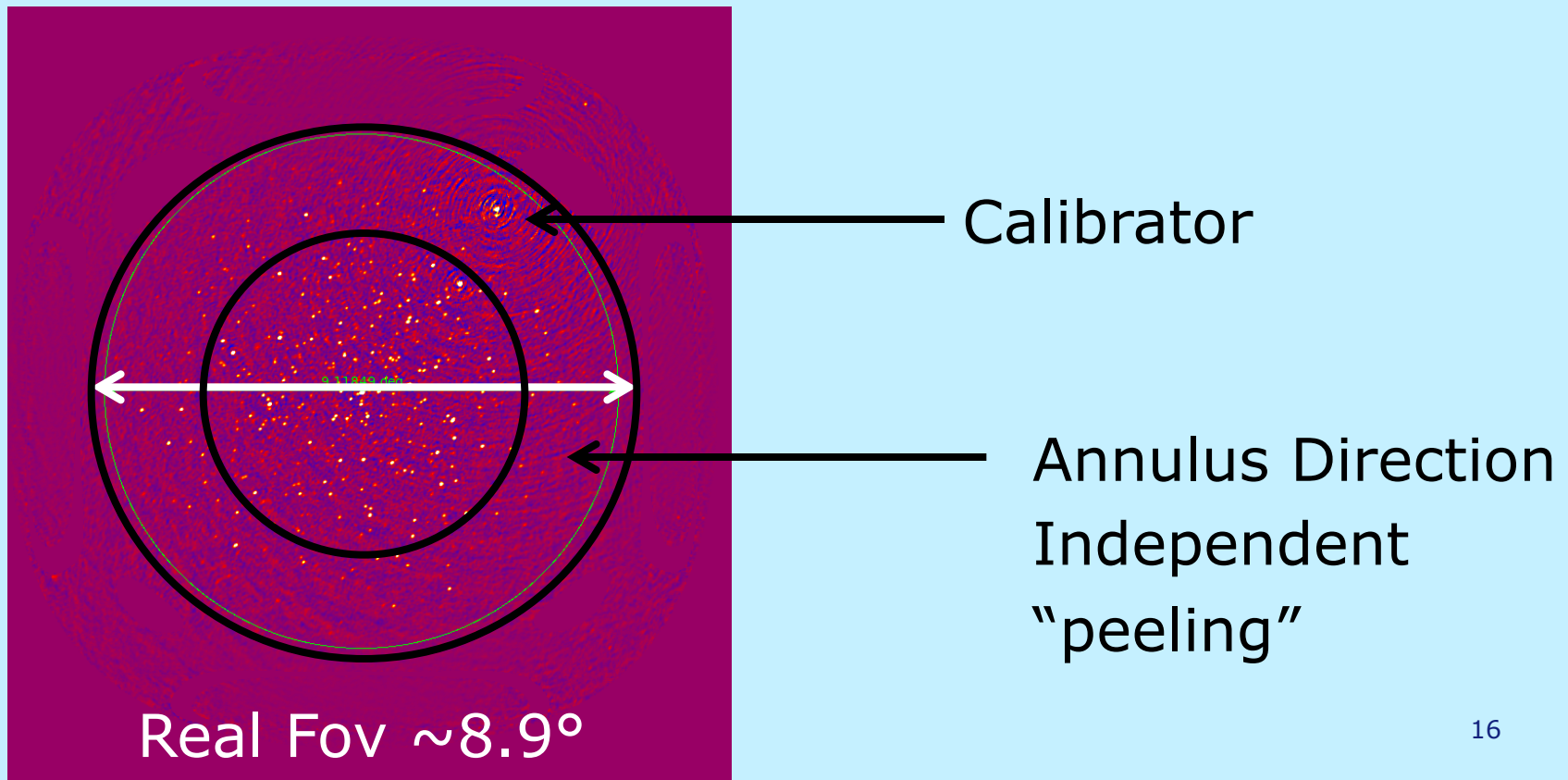
## Pre-Processing:

- **2 Preprocessing=> 2 Options ....**
  - Independent Direction Peeling (annulus subtraction)
  - Peeling (must be validated)

# Independent Direction Peeling (annulus subtraction)

**parameter investigation:** annulus cleaning:

outFOVclean=yes (radius selection)

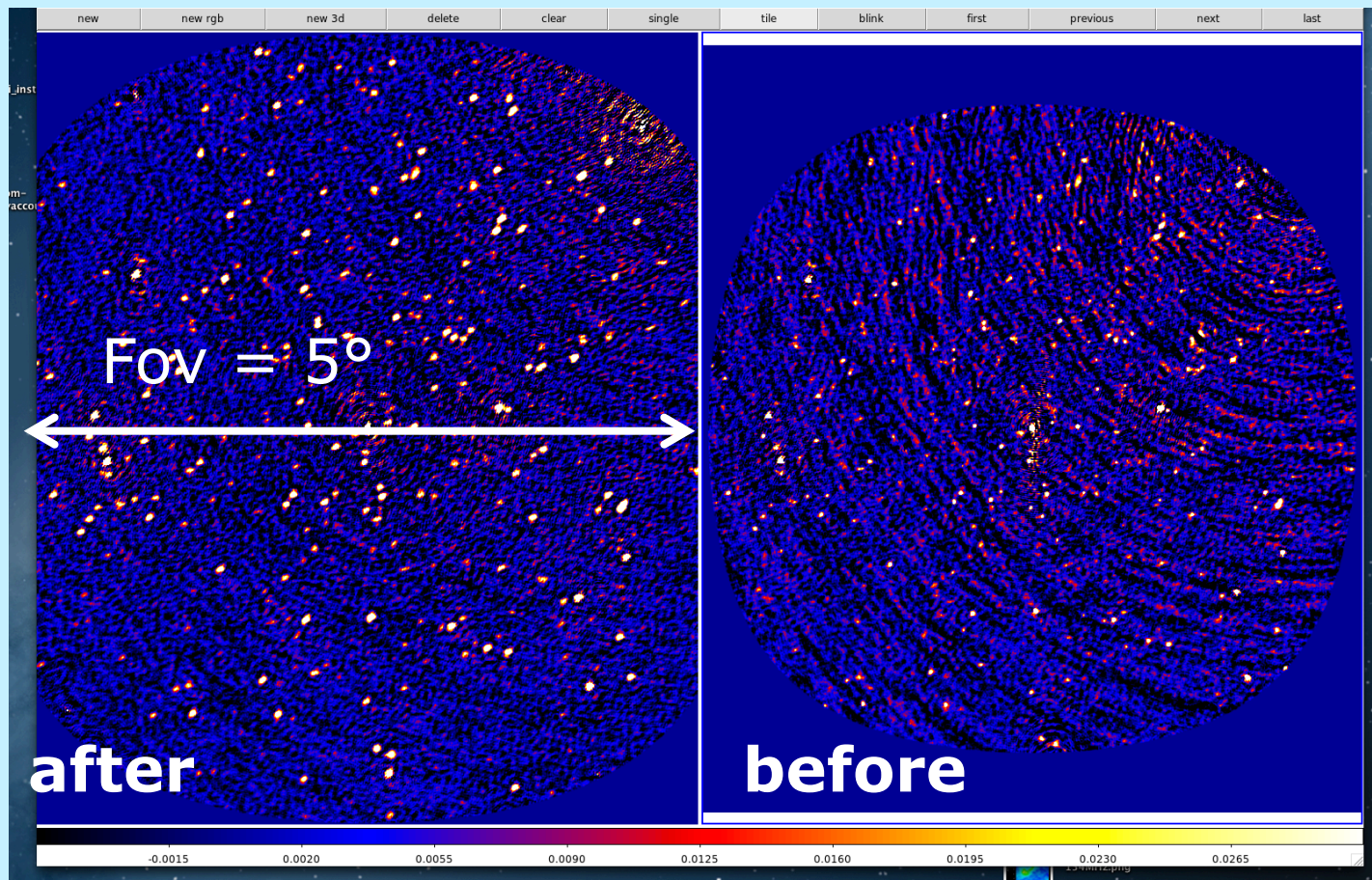


# Results on 3C388

New parameter validation: annulus cleaning:

Outfovclean

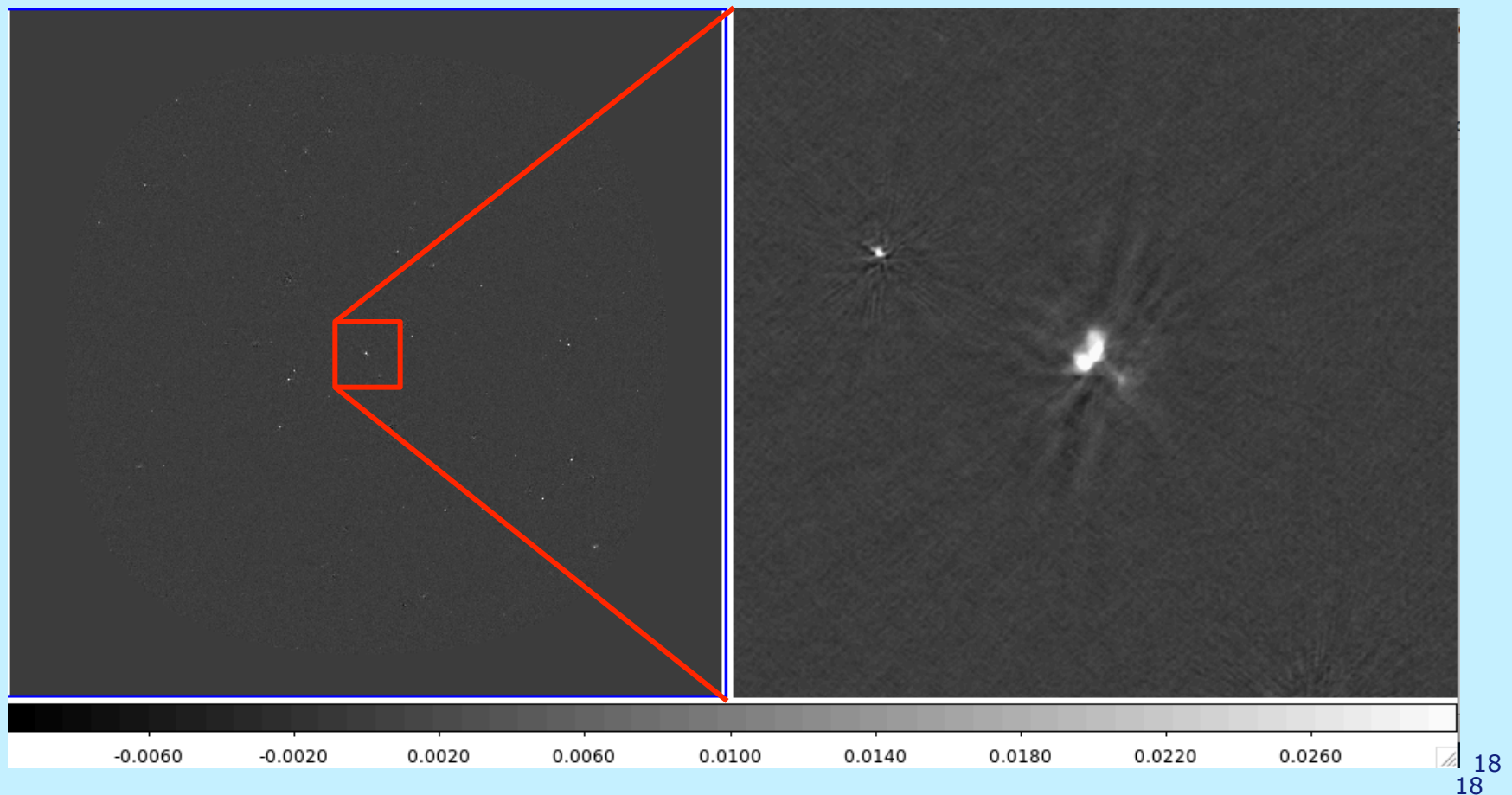
=yes (radius selection)



# Preliminary Results on J1431+1331 (280 subbands):

## **Complete study using 280 subbands (125-180 MHz):**

- Image on 280 Subbands: noise  $\sim 0.4\text{mJy}$   $\Rightarrow$  following expected statistics
- Thermal noise  $\sim 0.2\text{mJy}$   $\Rightarrow$  ratio  $\sim 2$



# Selfcal overview: peeling not tested yet ...

| Source Name | frequency  | best resol got? | resol | baseline | rms noise | Thermal noise | ratio    | Flux observed | Dynamic range | Flux overestimated   |
|-------------|------------|-----------------|-------|----------|-----------|---------------|----------|---------------|---------------|----------------------|
| 3C244       | 60MHz      | no              | 40"   | 10km     | 35mJy     | 3.6mJy        | 10.0     | 50Jy          | 1500          | ~0% (for the target) |
| 3C244       | 151MHz     | yes             | 5"    | 80km     | 1.35mJy   | 0.14mJy       | 10.0     | 22Jy          | 18500         | ~0%                  |
| J1431+1331  | 125-180MHz | yes             | 5"    | 80km     | 1.2mJy    | 0.2mJy        | 6.0      | 1.82Jy        | 2000          | ~30%                 |
| B2 0924+30  | 125-180MHz | no              | 10"   | ~15km    | 1.2mJy    | 0.2mJy        | 6.0      | diffuse       |               |                      |
| COMA        | 140MHz     | no              |       |          |           |               |          |               |               |                      |
| M82         | 127MHz     | yes             | 6"    | 80km     | 4.8mJy    | 0.85mJy       | 5.8      |               | 3500          |                      |
| M82         | 130MHz     | yes             | 6"    | 80km     | 4.3mJy    | 0.85mJy       | 5.05     |               | 3800          |                      |
| blank field | HBA        | no              | 25"   | ~20km    |           |               |          | 3Cxxx sources | high          |                      |
| H224+10     | 147.3MHz   | yes             | 48"   | ~10km    | 4.3mJy    | 1.12mJy       | 3.8      | 4.55          | 790           | ?                    |
| NGC3627     | 125.6MHz   | no              | 30"   | ~20km    | 10mJy     | 0.2mJy        | 50.0     | 2.71Jy        | 270           | ~30%                 |
| 3C452       | 146MHz     | yes             | 4.8"  | ~100km   | 2mJy      | 0.3mJy        | 6.0      | 71Jy          | 39250         | ~10%                 |
| 3C388       | 143MHz     | yes             | 4"    | ~120km   | 3mJy      | 0.2mJy        | 15.0     | 28Jy          | 9800          | ~5%                  |
| 3C48        | 116MHz     | yes             | 10"   | ~60km    | 6mJy      | 0.85mJy       | 7.0      | 71Jy          | 11200         | underestimated of 5% |
| M33         | 116MHz     | yes             | 10"   | ~60km    | 11mJy     | 0/85mJy       | 14.0     | diffuse       | 3800          |                      |
| empty field | 213MHz     | yes             | 3"    | ~120km   |           |               |          |               |               |                      |
| TOTAL       |            | 66%             |       |          |           |               | 11.5 / 8 |               |               |                      |

As an upper limit 66% of observations selfcal processed  
Have good results ... (less than 2 mJy noise)

Selfcal is in process : current conclusions ...

- Step dependent (in resolution) ... OK
- Use of clean components provide better results ... must be checked .... OK
- Use clean components with gaincal (Stefcal algorithm, salvini,2013) .... OK
- **Use model data column with casapy**
- **Awimager predict step**



# Parset example:

- `obsDir=/home/vilchez/DATA/3C35/chunk-SB100-SB119/DATA`
- `outputDir=/home/vilchez/DATA/3C35/chunk-SB100-SB119/selfcal_3cycle/  
no_propagation_300arcsec`
- `UVmin=0.0`
- `FOV=5`
- `resolutionVector=300,225,150,90,70,50,30`
- `robust=-0.3`
- `nofPixelPerBeam = 4`
  
- Then just type **`selfcal.py myparset`**